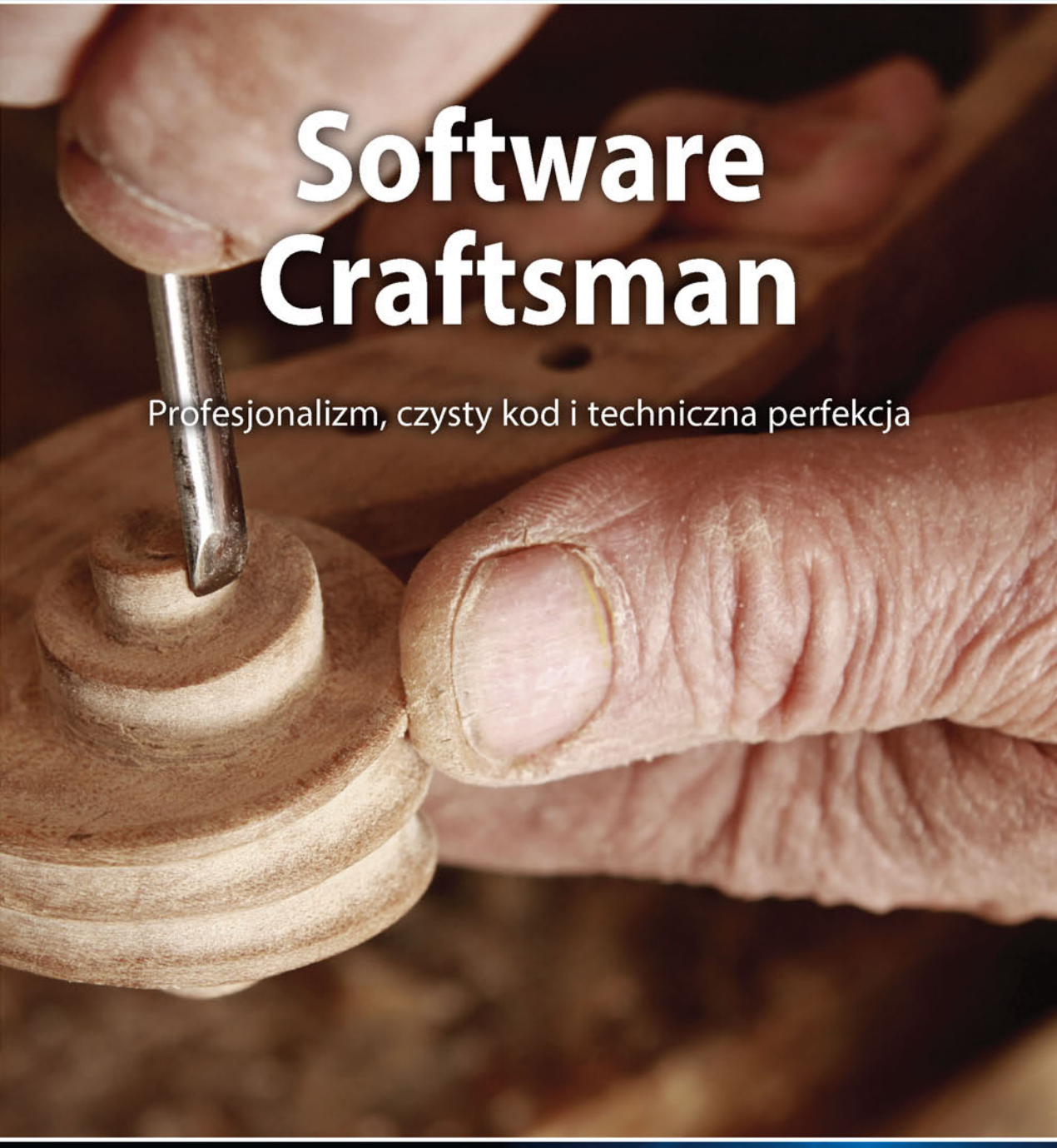
PRENTICE HALL

Helion



Software Craftsman

Profesjonalizm, czysty kod i techniczna perfekcja

Słowo wstępne Robert C. Martin

Sandro Mancuso

Tytuł oryginału: The Software Craftsman: Professionalism, Pragmatism, Pride

Tłumaczenie: Zbigniew Waśko

ISBN: 978-83-283-2135-9

Authorized translation from the English language edition, entitled: THE SOFTWARE CRAFTSMAN: PROFESSIONALISM, PRAGMATISM, PRIDE; ISBN 0134052501; by Sandro Mancuso; published by Pearson Education, Inc, publishing as Prentice Hall.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/prorze
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

5 BOHATEROWIE, ŻYCZLIWOŚĆ I PROFESJONALIZM

W latach 90. ubiegłego wieku zatrudniłem się w dużej międzynarodowej firmie. Po dwóch latach pracy w małych software house'ach i długim procesie rekrutacyjnym wydawało mi się, że wreszcie znalazłem pracę, o jakiej marzyłem. Tworzyliśmy system planowania zasobów przedsiębiorstwa (ERP), podobny do SAP ERP, a naszymi klientami były rządy i wielkie korporacje międzynarodowe. Zespół składał się z bardzo utalentowanych deweloperów, a sam projekt był niezwykle rozbudowany w porównaniu z tymi, nad którymi wcześniej pracowałem. Umiejętności obserwowane u kolegów z zespołu budziły we mnie kompleks niższości. Wszyscy wydawali mi się znacznie lepsi ode mnie, a ja koniecznie chciałem im dorównać. Pragnąłem za wszelką cenę pokazać, że też zasługuję na miejsce w tym zespole. Należałem do zespołu zajmującego się *architekturą* sytemu, a więc odpowiedzialnego za rozwijanie najbardziej fundamentalnych składników, infrastruktury komunikacyjnej i warstwy middleware, a także za generowanie kodu (przenoszenie kodu naturalnego do Delphi) i wspieranie zespołów *biznesowych* odpowiedzialnych za funkcje biznesowe systemu oraz interfejs użytkownika.

Projekt był naprawdę potężny. Na wspólnej bazie kodowej pracowało ponad stu deweloperów. Dużo kodu backendowego było tworzone w prawnie zastrzeżonych języku i bazie danych działających w środowisku mainframe, a oprogramowanie pośredniczące (ang. *middleware*) było pisane w C++. Jedyną rzeczą, którą cokolwiek

rozumiałem, było oprogramowanie klienckie pisane w Delphi. Wkrótce też przekonałem się, że nawet moja znajomość Delphi jest, delikatnie mówiąc, niewystarczająca.

Realizacja projektu zaczęła się dziesięć lat wcześniej i gdy przyszedłem, od dwóch lat trwały wytężone prace nad przenoszeniem dużej części systemu ze środowiska mainframe na platformę Windows. Zaczynały się też rozmowy nad przeniesieniem systemu do sieci, co było wielkim wyzwaniem dla wszystkich zespołów, ponieważ w większości składały się one z deweloperów pracujących wcześniej w środowisku mainframe, którzy dopiero zaczynali poznawać Delphi. Nie mieli oni też żadnego wzorca, z którym mogliby porównać własną aplikację, gdyż byłby to pierwszy duży projekt sieciowy w tej branży. Przygotowanie takiej migracji zlecono naszemu zespołowi.

Projekt był realizowany w klasycznym stylu kaskadowym i w ciągu tych dwóch i pół roku, jakie tam spędziłem, nigdy nie rozmawiałem z żadnym z naszych klientów. Żadnego z nich nawet nie widziałem. Sądzę, że ani jeden deweloper nie kontaktował się z klientami. Kierownictwo i jeszcze paru innych ludzi ciągle przydzielali nam nowe zadania i wyznaczali nieprzekraczalne terminy — nie miałem pojęcia, kim byli ci inni ludzie ani co robili. Nie miałem wprawdzie dostępu do prawdziwych liczb, ale z rozmów przy kawie dowiedziałem się, że kontrakty zawierały duże kwoty dotyczące premii i kar. Za każdym razem, gdy klient prosił o nową instalację lub zaktualizowanie oprogramowania, wszyscy wpadali w panikę.

Firma zatrudniała wielu deweloperów (w tym mnie) mieszkających około stu kilometrów od São Paulo (gdzie była siedziba firmy), do którego to miasta byliśmy dowożeni specjalnym autokarem. Musiałem więc wstawać przed 5, aby o 5.30 być już na miejscu zbiórki. Do domu wracałem zazwyczaj około 20, ale gdy zdarzały się nadgodziny, co nie było wcale rzadkością, autokar odjeżdżał beze mnie — musiałem wtedy korzystać z transportu publicznego i docierałem do domu już po północy.

Było coraz gorzej. Wyznaczono nam termin dostarczenia jednego z modułów systemu rządowi Izraela. Ze względu na różnicę stref czasowych i poziom desperacji, jaka w firmie zapanowała, musieliśmy pracować do późna. Skoro nie mogłem korzystać z autokaru, postanowiłem dojeżdżać codziennie samochodem, co zajmowało mi 90 minut w jedną stronę. Firma nie zwracała mi ani za paliwo, ani za opłaty drogowe, ponieważ zapewniała mi darmowy autokar. W ciągu dwóch miesięcy wielokrotnie pracowaliśmy przez 24 godziny bez większych przerw. Po takiej „dniówce" spędzenie 90 minut za kierownicą było zbyt niebezpieczne, więc spałem wtedy w samochodzie stojącym na firmowym parkingu. Trzy albo cztery razy pracowaliśmy nieprzerwanie przez 36 godzin, po czym kładliśmy się w samochodach na czas od 2 do 5 godzin, by potem znów siadać przy biurkach. Gdy koledzy chcieli mnie o coś zapytać, budzili mnie, stukając w szybę samochodu. Oczywiście nigdy nie otrzymaliśmy żadnej zapłaty za nadgodziny. W końcu po wielu rozmowach z zarządem zdecydowano się na opłacenie nam hotelu w pobliżu firmy.

Koledzy z zespołów biznesowych mówili, że to, co robimy, jest głupie — pracujemy na okrągło, śpimy w samochodach, a nikt nam nie płaci za nadgodziny. „Co wy wyprawiacie? Dlaczego się na coś takiego godzicie?" — pytali. Ja jednak nie widziałem w tym nic złego. Byłem samotny, żyłem na własną rękę i kochałem swoją pracę. Nade wszystko chciałem pokazać, na co mnie stać i że jesteśmy w stanie uratować projekt. Chcieliśmy być bohaterami. Dla mnie liczyło się tylko to, że dużo się uczę, pracuję z doświadczonymi deweloperami i robię to, co naprawdę kocham — piszę dobrej jakości kod. Koledzy z zespołu czuli podobnie. Byliśmy wyczerpani, ale w głębi ducha czuliśmy zadowolenie.

Gdy dzisiaj wspominam tamte czasy, widzę, jak szalone to było. Nasze postępowanie było kompletnie nieprofesjonalne. Nigdy nie pytaliśmy, dlaczego mamy robić to czy tamto. Nigdy nie próbowaliśmy dociekać, o co tak naprawdę chodzi klientowi, aby ewentualnie podpowiedzieć mu jakieś alternatywne rozwiązanie. Nie mieliśmy kontaktu z klientami i nie kwestionowaliśmy ich wymagań. Nigdy też nie mówiliśmy, że coś jest niemożliwe. Po prostu wykonywaliśmy swoją pracę, ponieważ wydawało nam się, że jesteśmy profesjonalistami.

W głębi duszy liczyliśmy także na odrobinę uznania i sławy. Marzyło nam się, że będziemy postrzegani jako ci, którzy uratowali projekt, a także jako ci, którzy dokonali rzeczy niemożliwej. Ostatecznie okazało się jednak, że cały nasz wysiłek był niepotrzebny. Termin został przesunięty i nikogo nie obchodziło, że tak ciężko pracowaliśmy. Nie staliśmy się sławni. Nadal byliśmy grupą kiepsko opłacanych deweloperów. Zdołaliśmy na czas wykonać to, co nam zlecono, ale z jakichś, nigdy niewyjaśnionych powodów termin został przesunięty. Sprzedawcy i ludzie odpowiedzialni za stronę biznesową przez cały ten czas pracowali w normalnym wymiarze, a czas wolny spędzali w domach z rodzinami. Tylko my, deweloperzy, pracowaliśmy jak osły, aby zadowolić ludzi, których nawet nie znaliśmy. Przy odgórnym zarządzaniu i systemie nakazowo-kontrolnym większość decyzji po prostu do nas nie docierała. Byliśmy jak robotnicy w fabryce. A jednak nas to cieszyło. Tworzyliśmy naprawdę fantastyczny zespół, a praca wśród wielu utalentowanych i rozumiejących się ludzi sprawiała nam wiele przyjemności.

Dzisiaj wiem, że to, co się wtedy działo, było złe. Boli mnie, gdy sobie przypominam, jak nas traktowano. Boli mnie, gdy sobie uświadamiam, że takie sytuacje są wciąż na porządku dziennym, szczególnie w krajach rozwijających się. Czuję niesmak na myśl o naszym nieprofesjonalnym działaniu — nigdy nie dociekaliśmy, dlaczego mamy robić to czy tamto ani na czym polega prawdziwy problem. Nigdy nawet nie próbowaliśmy szukać alternatywnych rozwiązań. Pozwalaliśmy na to, by nas traktowano jak robotników w fabryce. Zachowywaliśmy się jak robotnicy i to nas cieszyło. A tak naprawdę wyrządzaliśmy krzywdę sobie i firmie. Podczas tych długich godzin harówki popełnialiśmy mnóstwo błędów, włącznie ze skasowaniem o 3 nad ranem całej produkcyjnej bazy danych. Tworzyliśmy rzeczy zupełnie niepotrzebne. Nie wiedzieliśmy,

na czym polegają prawdziwe problemy, i nie stwarzaliśmy sytuacji, które mogłyby nas naprowadzić na właściwe rozwiązania. Pozwoliliśmy na zabranie sobie życia prywatnego, a nawet pogardzaliśmy tymi, którzy nie spali w samochodach lub pracowali *tylko* 12 godzin dziennie. Nie byliśmy profesjonalistami, ponieważ nigdy nie powiedzieliśmy „NIE".

NAUKA MÓWIENIA „NIE"

Bardzo często kierownictwo narzuca nam sztywne terminy i mocno naciska, byśmy ich dotrzymali. Często są to terminy nierealne, a kierownicy w sposób agresywny starają się wymusić na deweloperach wykonanie zadania na czas. Deweloperzy, zwłaszcza ci młodzi, z reguły ulegają presji i mówią, że zrobią wszystko w wyznaczonym terminie, gdyż boją się sprzeciwić kierownictwu. Mimo że w głębi ducha wiedzą, iż wykonanie zadania w tak krótkim czasie jest prawie niemożliwe, zgadzają się i podejmują zobowiązanie.

Rezultaty są zazwyczaj opłakane. Systemy wdrażane do produkcji są pełne błędów, klienci są niezadowoleni, a firma traci w ich oczach zaufanie. Nawet najcięższa praca po kilkanaście godzin dziennie kosztem czasu, który można by spędzić z najbliższymi, nie zagwarantuje, że system będzie działał poprawnie. A ten sam kierownik, który przydzielił deweloperom zadanie praktycznie niewykonalne, podczas gdy sam spędził weekend w domu z rodziną, będzie im zarzucał brak kompetencji. „Dlaczego u licha przerwaliście walidację na stronie rejestracji? Dlaczego system nie działa prawidłowo? Jak to możliwe, że system nie jest w stanie przetworzyć więcej niż 10 000 transakcji na minutę? Jak można było zapomnieć o tym skrajnym przypadku?".

KLĘSKA

Pracowałem kiedyś w dużej firmie telekomunikacyjnej. Zespołowi marketingowemu spieszyło się z otwarciem nowego portalu mobilnego, nad którym właśnie pracowaliśmy. Chciano go uruchomić w trzech krajach jednocześnie, a zapewniano nas, że w pierwszych dniach tylko niewielka część klientów będzie używała nowej aplikacji. Łączna liczba klientów w tych trzech krajach wynosiła prawie 20 milionów, więc — mówiąc o niewielkiej części — miano na myśli kilkaset tysięcy osób. My takiej wiedzy nie posiadaliśmy. Kierownik przyszedł do nas i powiedział, że chce, aby aplikacja była gotowa przed takim a takim dniem. Chociaż mieliśmy przed sobą kilka miesięcy, byliśmy niemal pewni, że nie zdążymy zrobić wszystkiego, czego żądał. Na jednym z pierwszych spotkań zasygnalizowaliśmy swoje wątpliwości, ale w odpowiedzi usłyszeliśmy tylko, że zadanie musimy wykonać i że jeśli weźmiemy się solidnie do pracy, to na pewno damy radę. Gdy jeszcze próbowaliśmy oponować, dodał, że nie ma innej opcji, więc nie pozostało nam nic innego, jak powiedzieć, że *zrobimy wszystko, co w naszej mocy*.

Tydzień po tygodniu kierownik i analityk biznesowy poszerzali aplikację o kolejne funkcje, ale terminu zakończenia prac nie przesuwali. Wielokrotnie mówiliśmy im, że ledwie dajemy sobie radę z istniejącymi już wymaganiami, więc z nowymi na pewno nie zdążymy. Wracali jednak i stwierdzali, że zespół marketingowy rozpoczął już kampanię reklamową we wszystkich trzech krajach i informacje o nowych funkcjach zostały już podane.

Twierdziliśmy, że zespół marketingowy powinien najpierw porozmawiać z nami i wspólnie powinniśmy ustalić, które funkcje rzeczywiście muszą być zaimplementowane w całości, a które można by zaimplementować tylko częściowo. Mieliśmy kilka pomysłów, dzięki którym niektóre funkcje udałoby się zdecydowanie zmniejszyć albo nawet wyeliminować. Kierownik odparł, że to on jest naszą skrzynką kontaktową i zna wszystkie wymagania marketingowców, więc nie ma potrzeby, byśmy z nimi rozmawiali. Za każdym razem powtarzał: „Niestety, nie ma miejsca na negocjacje. Musicie zaimplementować wszystko”. Każda rozmowa z nim kończyła się tym, że nie mamy wyjścia i musimy pracować tak ciężko, jak tylko potrafimy. Co jeszcze mogliśmy zrobić?

Przez kilka tygodni pracowaliśmy również po godzinach i w weekendy. Oczywiście mam na myśli tylko deweloperów. Kierownik z analitykiem biznesowym znikali zaraz po 18, a w weekendy nawet nie odbierali telefonów. Od samego początku mówiliśmy, że trzeba przygotować określoną infrastrukturę dla prac nad tym projektem, że musimy mieć kwaziprodukcyjne środowisko do testowania aplikacji. Zamiast uzyskać zgodę na wykonanie czegoś takiego, słyszeliśmy, że mamy się skupić na dopracowaniu funkcji biznesowych, ponieważ zespół marketingowy chce opublikować wersję demonstracyjną aplikacji. Na niecały miesiąc przed ostatecznym terminem oznajmiliśmy, że musimy przerwać prace nad funkcjami biznesowymi i zająć się technicznymi aspektami całości, aby zagwarantować prawidłowość działania aplikacji, i w związku z tym nie zdążymy opracować niektórych funkcji. Odpowiedź znów brzmiała *nie*. Mieliśmy skończyć wszystkie funkcje biznesowe. „Nie kombinujcie, to już końcówka, wiem, że dacie radę” — usłyszeliśmy od kierownika. Jeszcze raz poprosiliśmy o możliwość porozmawiania z marketingowcami, z którymi nie spotkaliśmy się od momentu rozpoczęcia prac nad projektem, lecz i tym razem odpowiedź była negatywna. W obliczu zbliżającego się terminu zakończenia prac i ze świadomością, ile jeszcze pracy zostało do wykonania, mogliśmy odpowiedzieć tylko: „Oczywiście, postaramy się”.

W końcu nadszedł ten dzień. Był poniedziałkowy poranek, gdy aplikacja ujrzała światło dzienne. Prace produkcyjne zakończyły się zaledwie kilka godzin wcześniej. Byliśmy zmęczeni i chcieliśmy, żeby to się już skończyło, ale z drugiej strony byliśmy niezwykle dumni z siebie, gdyż wierzyliśmy, że dzięki nam projekt został uratowany. Czekaliśmy na słowa uznania. Czuliśmy się jak bohaterowie. W środku nocy, podczas przerwy na kawę, rozmawialiśmy nawet o czekających nas podwyżkach i awansach.

System wystartował, ale po niecałej godzinie już nie działał. Obciążenie było zbyt duże. Po trzech godzinach udało nam się przywrócić go do życia, ale znów tylko na parę godzin. Sytuacja powtórzyła się jeszcze wiele razy w następnych pięciu dniach. Wszyscy, włącznie z marketingowcami (wreszcie ich zobaczyliśmy), zaglądaliśmy sobie przez ramię, pytając, co się dzieje z tym systemem. Testy? Nie. Było tylko kilka. Kto miał czas na pisanie testów? Mimo że byliśmy małym zespołem, stworzyliśmy kod niezwykle chaotyczny, w którym wprowadzanie zmian było niezwykle trudne, podobnie zresztą jak testowanie — niemal wszystko trzeba było sprawdzać ręcznie.

Podsumowując, firma została zasypana skargami i pretensjami klientów. Po kilku tygodniach udało nam się pokonać wszystkie problemy i ustabilizować system, ale oczywiście zamiast premii i awansów zyskaliśmy tylko złą reputację. Kierownik tłumaczył się przed zespołem marketingowym, że to my, deweloperzy, twierdziliśmy, że wszystko da się zrobić, a on polegał na naszej opinii i wierzył w nasze umiejętności. Przecież on nie pisał kodu, prawda? Więc to nie jego wina.

Gdy już było po wszystkim, marketingowcy powiedzieli nam, że gdyby wiedzieli o naszych kłopotach i wątpliwościach co do wydolności systemu, zrezygnowaliby z kilku funkcji. Zgłaszali zapotrzebowanie na coraz więcej rzeczy, ponieważ sądzili, że jesteśmy w stanie je zaimplementować i że wszystkie pozafunkcjonalne wymagania mamy pod kontrolą. Nikt im nie powiedział, że jest inaczej.

Ostatecznie większość z nas rozstała się z firmą, co zmusiło jej zarząd do szybkiego zatrudnienia nowych deweloperów. Firma nie miała czasu na wybranie najlepszych kandydatów, więc szybko wpadła w jeszcze większe tarapaty, gdyż nowi deweloperzy nie mieli pojęcia o tych wszystkich pułapkach, jakie na nich czyhały w pozostawionym przez nas kodzie. Zapowiedziano wypuszczenie drugiej wersji aplikacji w ciągu trzech miesięcy, ale z tego, co słyszałem, udało im się to dopiero po trzech kwartałach.

WNIOSKI

W powyższej historii od razu widać, że kierownik nie działał profesjonalnie. Nie dbał o zespół i myślał tylko o własnym awansie, podczas gdy całą pracę wykonywali za niego inni. Aż ciśnie się na usta, że to on zawinił i on powinien być zwolniony oraz że deweloperzy powinni się bronić i opowiedzieć wszystkim, jak się kierownik zachowywał i jak oni ciężko pracowali, aby ratować projekt. Owszem, możemy to wszystko powiedzieć, ale to już niczego nie zmieni. Teraz jest już za późno.

Całej tej sytuacji można było uniknąć, a winą za niepowodzenie powinno się obarczyć nas, deweloperów. My też nie działaliśmy profesjonalnie. Nie powinniśmy mówić *spróbujemy*, jeśli wiedzieliśmy, że zadanie jest niewykonalne. Nawet jeśli wierzyliśmy, że możliwe jest zaimplementowanie wszystkich funkcji, powinniśmy zdawać sobie

sprawę, że nie zdołamy wszystkiego przetestować, a więc nie będzie żadnej pewności co do sprawności aplikacji. Niezależnie od niechęci do przeciwstawiania się kierownikowi było w nas coś, co nakazywało nam godzić się na jego warunki. Gdzieś w głębi odczuwaliśmy pragnienie pokazania światu, jacy jesteśmy dobrzy. Mimo że wiedzieliśmy, jak trudne to będzie, to jednak cały czas liczyliśmy, że jest niewielka szansa na sukces, a jeśli się uda, będziemy bohaterami. Będą o nas mówić jako o tych, dzięki którym udała się wielka kampania reklamowa. Gdzieś w głębi liczyliśmy, że tak się stanie.

W tej sytuacji nie powinniśmy nigdy mówić, że spróbujemy, ani myśleć o bohaterstwie. Od początku powinniśmy naciskać na kierownika w sprawie kontaktów z działem marketingu. Powinniśmy się zorientować, że kierownik nie mówi wszystkiego marketingowcom, skoro ciągle dodaje nam nowe funkcje, wiedząc, że nasze możliwości są ograniczone. Powinniśmy dokumentować przebieg każdego spotkania, podczas którego informowaliśmy kierownika o istniejących ograniczeniach i potrzebie przygotowania systemu do obsłużenia dużej liczby klientów. Kierownik, choć robił to fatalnie, wykonywał swoją pracę — pilnował, aby aplikacja zawierała wszystkie funkcje, jakich żądali marketingowcy. My powinniśmy pracować lepiej i wykazując więcej realizmu, nie przyjmować na siebie odpowiedzialności za wykonanie czegoś, co — o czym wiedzieliśmy — będzie kiepskiej jakości i nie będzie się nadawało do wdrożenia. Gdy zaczęto dodawać nowe funkcje, powinniśmy stanowczo zażądać spotkania z zespołem marketingowym. Powinniśmy wyraźnie powiedzieć, że system pełen nawet najlepszych funkcji będzie zupełnie bezużyteczny, jeśli będzie działał źle lub wcale. Myślenie o bohaterstwie sprawiło, że nie byliśmy profesjonalistami. Nie umieliśmy powiedzieć *nie*.

PROFESJONALIZM

Trudne do dotrzymania terminy zdarzają się nam bardzo często. Najlepszym sposobem radzenia sobie z nimi jest przeanalizowanie wszystkiego, co trzeba zrobić, oraz poinformowanie o wszelkich zagrożeniach, jakie dostrzegamy, a także o obawach, jakie mamy. Wszystkie obawy i niejasności należy zgłaszać najszybciej jak to możliwe. Powinniśmy też określić, co jesteśmy w stanie wykonać należycie przed wyznaczonym terminem. Przez wykonanie należyte rozumiem napisanie kodu i wszechstronne przetestowanie go w środowisku podobnym do produkcyjnego.

Kierownicy często nie traktują poważnie słów deweloperów, gdy ci mówią, że nie są w stanie wykonać należycie wszystkiego przed wyznaczonym terminem,. Zaczynają się wtedy negocjacje, ale niemający umiejętności negocjacyjnych i dobrze przygotowanych argumentów deweloperzy najczęściej ulegają presji i biorą na swoje barki więcej, niż mogą udźwignąć. Niektórzy kierownicy bywają bardzo przekonujący w namawianiu do przyjęcia terminu, z którego trudno się wywiązać. „Chłopaki, wiecie, jakie to jest ważne. Zdaję sobie sprawę, że nie będzie łatwo, ale wierzę w was. To jest naprawdę

potrzebne. Jestem pewien, że jeśli się przyłożycie, to dacie radę". Aby uniknąć konfrontacji lub zademonstrować ochotę do pracy, deweloperzy odpowiadają: „OK. Damy z siebie wszystko".

Gdy mówimy, że spróbujemy, zwykle druga strona odbiera to jako naszą gotowość do wykonania zadania. Nasze *spróbuję* jest równoznaczne z *tak, będzie zrobione.* Brzmi to również jak wyznanie, że na co dzień nie pracujemy zbyt ciężko. Zupełnie jak byśmy powiedzieli, że mamy spory zasób energii i w każdej chwili możemy go wykorzystać, aby przyspieszyć pracę.

Skoro wiemy, że wykonanie zadania w terminie jest praktycznie niemożliwe, to dlaczego mówimy kierownictwu, że spróbujemy je wykonać? Co to znaczy *spróbujemy*? Czy to znaczy, że jesteśmy w stanie wykonać każde zadanie, jeśli tylko będziemy pracować intensywniej? Czy to znaczy, że zobowiązujemy się pracować po godzinach i w weekendy? Jeśli tak, to czy to coś zmienia? Jeżeli nie, wówczas oznacza to, że albo na co dzień nie pracujemy zbyt intensywnie, albo po prostu kłamiemy.

W swojej książce *The Passionate Programmer* Chad Fowler twierdzi, że mówienie *tak* tylko dlatego, by nie zawieść drugiej strony, jest zwykłym kłamstwem. Idzie nawet dalej, utrzymując, że mówienie *tak* jest nawykiem uzależniającym i wyniszczającym. Jest to zły nawyk, który tylko pozornie wygląda na dobry.

Z reguły nie lubimy mówić *nie*. To źle brzmi. Gdy mówimy *nie*, czujemy się jak przegrani. Wydaje nam się, że jesteśmy gorsi od innych i rozczarowaliśmy kolegów z zespołu. Chcemy odnieść sukces, ale musi to być pewny strzał. Chociaż wydaje się to odruchem pozytywnym, jest też przejawem egoizmu. Należy pamiętać, że gdy mówimy *tak*, inni mogą na tej podstawie coś zaplanować. Nasi przełożeni mogą przedstawić plany swoim przełożonym, innym zespołom, klientom lub partnerom biznesowym. Wypowiadanie się w sposób nierzetelny i niejednoznaczny może wyrządzić wiele szkód całej firmie. Profesjonalizm oznacza bycie uczciwym wobec siebie, swojego zespołu, kierownictwa i klientów.

Czasami czujemy, że nasz kierownik nie mówi swojemu szefowi — lub komukolwiek, kto finansuje projekt — wszystkiego o bieżącym stanie projektu. Wyczuwamy, że nie informuje innych ludzi o naszych obawach, lecz zawsze mówi im, iż wszystko jest w porządku. Powinniśmy go wtedy spytać, dlaczego tak robi. Jeśli sytuacja jest naprawdę zła, powinniśmy go uprzedzić, że rozdmuchamy sprawę i gdy o wszystkim dowie się jego szef, poprosimy o spotkanie w celu przedyskutowania kroków zaradczych. Może brzmi to groźnie i agresywnie, ale będzie z korzyścią dla zespołu, firmy i nierzadko także dla naszego kierownika. Sztuki negocjowania powinniśmy się uczyć wszyscy. Tak jak kierownictwo wykonuje swoją pracę, my powinniśmy wykonywać swoją — podejmując trudne decyzje i nie unikając konfrontacji. Jeśli robimy to w sposób uczciwy i transparentny, jest duża szansa, że nikt nie ucierpi, a zespół i firma zyskają.

WYSUWANIE PROPOZYCJI

Mówienie zawsze *nie* też nie jest podejściem profesjonalnym. Po każdym *nie* powinna następować lista rozwiązań alternatywnych. Zanim to słowo wypowiemy, powinniśmy przeanalizować problem i zaproponować jakieś wyjście z sytuacji. Nie zawsze takowe znajdziemy, ale przynajmniej powinniśmy próbować. Czasami nawet niedopracowany pomysł może pomóc w znalezieniu dobrego rozwiązania. Ostatecznie wszyscy powinniśmy uczestniczyć w rozwiązywaniu problemów i powinno być wybrane rozwiązanie najlepsze, bez względu na to, kto je zaproponował.

Mówienie *nie* bez proponowania czegokolwiek w zamian w niczym nie pomaga, ponieważ druga strona niewiele może z taką naszą odpowiedzią zrobić. Pamiętam zasłyszaną kiedyś opowieść o dyrektorze, który właśnie objął stanowisko w nowym dla siebie mieście i poprosił dwie asystentki, aby mu kupiły sok pomarańczowy. Pierwsza pobiegła do sklepu za rogiem, zapytała o sok i po 5 minutach wróciła z informacją, że w sklepie nie mają ani soku, ani pomarańczy. Druga wróciła dopiero po 20 minutach i zanim zdążyła cokolwiek powiedzieć, dyrektor zapytał: „Już wiem, że w sklepie za rogiem nie ma soku pomarańczowego, dlaczego więc tak długo cię nie było?". „Tak" — odparła. „W sklepie nie mają soku pomarańczowego, ale mają ananasowy i jabłkowy. Ananasowy jest świeżo wyciskany, a jabłkowy mają butelkowany. Dowiedziałam się również, że trochę dalej jest supermarket, gdzie można kupić pomarańcze. W ciągu pięciu minut może pan mieć sok ananasowy lub jabłkowy albo w ciągu pół godziny zrobię świeży sok pomarańczowy. Muszę tylko wiedzieć, co pan wybiera".

Pierwsza asystentka, chociaż wróciła szybko, pozostawiła swojego szefa z niczym. Musiał sam rozwiązywać problem, nie znając w ogóle okolicy. Druga asystentka zaproponowała mu kilka opcji, z których mógł wybrać według niego najlepszą. Dobre chęci zawsze się liczą, nawet więc gdy mówimy *nie*, zawsze powinniśmy dążyć do tego, by powiedzieć *tak*.

Czasami są jednak problemy, których po prostu nie umiemy rozwiązać. W takich sytuacjach powinniśmy się do tego uczciwie przyznać, i to najszybciej, jak się tylko da. Należy też okazać gotowość do przyjrzenia się problemowi i dołożenia wszelkich starań w celu jego rozwiązania, ale z wyraźnym zaznaczeniem, że nie obiecujemy nic więcej poza dzieleniem się swoją wiedzą. Najlepsze, co możemy zrobić, to jak najczęściej informować o swoich postępach; umożliwi to zespołowi podejmowanie decyzji, co robić z każdą nową porcją informacji. Dzięki nowej informacji zespół może rozwinąć inne pomysły, a także zaoferować nam pomoc w obszarach, w których nie czujemy się zbyt pewnie. Takie zachowanie sprawi, że nikt w zespole nie będzie wątpił w nasze słowa, gdy stwierdzimy, że coś wiemy. Bycie uczciwym nawet w najtrudniejszej sytuacji jest oznaką profesjonalizmu.

OPCJA NIEOCZEKIWANA I REALNA

Całkiem niedawno pracowałem nad projektem dla banku inwestycyjnego. Projekt dotyczył raportowania transakcji dla instytucji nadzoru bankowego. Termin zakończenia prac został narzucony przez wspomnianą instytucję, więc w banku nie było z kim rozmawiać na ten temat. Sankcją za niedotrzymanie terminu mogła być wysoka kara pieniężna lub zakaz sprzedaży pewnych produktów, co oznaczałoby poważne naruszenie reputacji banku.

Mieliśmy bardzo zdolny zespół i pracowaliśmy ciężko, żeby zdążyć. Nadszedł jednak moment, w którym stało się jasne, że nie potrafimy spełnić wszystkich obowiązkowych wymagań — mieliśmy problem z automatyzacją procesu raportowania wszystkich typów transakcji wymaganych przez nadzór.

Kierownictwo naciskało, żebyśmy pracowali po godzinach i w weekendy, gdyż stawka była naprawdę wielka. Wszystkie banki inwestycyjne w Wielkiej Brytanii przechodziły przez to samo, a o nowych przepisach mówiło się we wszystkich wiadomościach finansowych. Zarząd wciąż powtarzał, że negocjowanie warunków projektu lub terminu nie wchodzi w grę, gdyż nikt nie może w tej sprawie nic zrobić. Albo raportowanie będzie działało, albo zapłacimy kary i stracimy dobrą reputację.

W wyniku wymagań instytucji nadzorującej wyznaczano różne terminy wprowadzania raportowania transakcji spełniających określone kryteria. Było jasne, że nasza aplikacja będzie się rozwijać, a zatem nie mogliśmy sobie pozwolić na rezygnację z przyjętych standardów i praktyk kodowania tylko po to, aby zdążyć przed pierwszym terminem. Spowodowałoby to obniżenie jakości naszej bazy kodowej, a to z kolei znacznie utrudniłoby dotrzymanie terminów dla pozostałych typów transakcji.

Zgodziliśmy się na wydłużenie dnia pracy w tygodniu, ale odmówiliśmy pracy w weekendy, a także odrzuciliśmy propozycję zarządu dotyczącą poszerzenia składu zespołu. Uznaliśmy, że sytuacja tylko by się przez to pogorszyła. Nadal składaliśmy zarządowi codzienny raport z postępu naszych prac i ciągle pokazywaliśmy, że pierwszy termin jest zagrożony. Wystarczyło zresztą rzucić okiem na naszą tablicę kanban, aby zobaczyć, na jakim etapie jesteśmy. Po dwóch czy trzech tygodniach oznajmiliśmy, że nie będziemy już pracować po godzinach.

Nie muszę chyba mówić, jaką to wywołało panikę i strach przed konsekwencjami niedotrzymania terminu. Wytrwanie we własnym postanowieniu i niepoddanie się presji kierownictwa były dla nas niezwykle trudne. Ale wytrzymaliśmy. Wiedzieliśmy, że obniżenie standardów i pracowanie przez siedem dni w tygodniu spowoduje, że się wypalimy i realizacja projektu jako całości będzie zagrożona.

Na szczęście pracowaliśmy zgodnie z zasadami Agile i na białej tablicy zawsze był widoczny stan naszych prac oraz to wszystko, co nas blokowało lub było wąskim gardłem. Każdy z nas miał też ciągły dostęp do rejestru produktu. Wizualizacje tego, co już było zrobione, i tego, co należało jeszcze zrobić, pomagały zarządowi w orientowaniu się, jaka jest aktualna sytuacja, a nam pomagały w przekonywaniu, że terminu nie da się dotrzymać, nawet jeśli będziemy pracować przez siedem dni w tygodniu.

Powiedzenie *nie* już pierwszego dnia i nieukrywanie niczego przed stroną biznesową daje poczucie wolności. Każdy ma wtedy swobodę w poszukiwaniu innych realnych rozwiązań. Okazało się, że istniały dwa sposoby dostarczania raportów do instytucji nadzorującej. Pierwszy miał polegać na pobieraniu informacji generowanych przez system administracyjny, analizowaniu ich, uzupełnianiu, przekształcaniu i wysyłaniu do instytucji nadzorującej w sposób całkowicie zautomatyzowany — i to właśnie miał robić system, nad którym pracowaliśmy. Drugi sposób to ręczne wysyłanie wygenerowanych przez banki plików CSV za pośrednictwem interfejsu dostarczonego przez instytucję nadzorującą. Ze względu na dużą liczbę transakcji uznano już na początku (jeszcze przed rozpoczęciem prac nad naszym projektem), że drugi sposób jest praktycznie niewykonalny i odrzucono go.

Pracowaliśmy nad pierwszą opcją, ale skoro już wiedzieliśmy, że system nie będzie ukończony na czas dla wszystkich typów transakcji, jakie miały być raportowane, zaczęliśmy analizować wykonalność metody ręcznej. Na początku nikt tego nie brał pod uwagę, ale teraz sytuacja wyglądała inaczej. Raportowanie większości typów transakcji udało nam się już zautomatyzować i na raportowanie ręczne przypadało stosunkowo niewiele pracy. Przekalkulowaliśmy koszty ręcznego przetwarzania i wysyłania raportów przez oddelegowanych (lub wynajętych) do tej pracy ludzi i porównaliśmy je z ceną, jaką przyszłoby zapłacić za niedotrzymanie terminu. Okazało się, że pierwsza opcja jest z biznesowego punktu widzenia znacznie lepsza. Podjęto więc stosowne decyzje i, pomijając kilka drobnych problemów, które jeszcze trzeba było rozwiązać, pierwsze raporty zostały wysłane w terminie.

Gdybyśmy od razu nie powiedzieli *nie* i ukrywali prawdę przed zarządem, licząc, że jakoś nam się uda i zostaniemy bohaterami, prawdopodobnie nasze wysiłki zakończyłyby się niepowodzeniem, a firma poniosłaby wielkie straty. Powiedzenie *nie* uwolniło nas od myślenia wyłącznie o zbliżającym się terminie i pozwoliło skupić się na poszukiwaniu innych rozwiązań zaistniałego problemu.

Opisane wyżej połączenie metod ręcznej i automatycznej, chociaż nie było rozwiązaniem idealnym, to jednak pozwoliło nam na spokojne kontynuowanie prac przy jednoczesnym spełnieniu wszystkich wymagań instytucji nadzorującej.

Zyskaliśmy czas na zautomatyzowanie całego procesu raportowania i wcale nie musieliśmy rezygnować z naszych standardów kodowania. Co więcej, mogliśmy stopniowo uruchamiać kolejne etapy automatyzacji, co pozwalało na dokładne sprawdzenie, czy wszystko działa zgodnie z oczekiwaniami.

OŚWIECONE KIEROWNICTWO

Dobre kierownictwo rozumie, że jest częścią zespołu i musi współpracować z deweloperami w dążeniu do osiągnięcia wspólnego celu. Dobre kierownictwo nie wprowadza podziału na „my" i „oni". Powinno znać bieżący stan projektu i wspólnie z zespołem ustalać, co będzie możliwe do wykonania w następnym przedziale czasowym.

Kierownictwo powinno doceniać naszą postawę, gdy mówimy, że pewnych rzeczy nie będziemy w stanie wykonać. Chociaż nie jest to dobra informacja, to jednak pozwala uniknąć znacznie większych problemów. Szczerość wzajemnych relacji umożliwia zespołowi i kierownictwu przygotowanie się do trudnych sytuacji. Zapalanie czerwonej lampki najwcześniej, jak to tylko możliwe, w celu zasygnalizowania, że coś jest nie tak, pozwala zespołowi i wszystkim zainteresowanym na zachowanie się w sposób pragmatyczny i podjęcie prób rozwiązania problemu. Ludzie uwolnieni od balastu niewykonalnego zadania stają się bardziej kreatywni i lepiej sobie radzą z problemami. Jeśli wywiązujemy się z powierzonych nam zadań, gdy mówimy *tak*, to kierownictwo będzie dużo bardziej skłonne zaufać nam również wtedy, gdy powiemy *nie*.

Mądrzy ludzie wiedzą, że nie wiedzą wszystkiego i że w grze drużynowej liczą się wszyscy członkowie zespołu. Jak głosi mądrość ludowa: łańcuch jest tak mocny, jak jego najsłabsze ogniwo. Mądre kierownictwo rozumie, że pewne rzeczy mogą być trudne do wykonania i trzeba na to więcej czasu. Docenia, gdy twardo mówimy, w czym rzecz, i jednocześnie sugerujemy inne rozwiązania. Takie podejście pokazuje, że wszyscy tworzą jedną drużynę i dążą do tego samego celu. Mając więcej informacji i kilka propozycji wyjścia z trudnej sytuacji, kierownictwo pewniej podejmuje decyzje; z pewnością też bardziej sobie ceni ludzi, którzy te propozycje zgłaszają.

Dobre kierownictwo jest częścią zespołu. Jest z nim na dobre i na złe. Jeśli wszyscy zostają po godzinach, kierownicy powinni zrobić to samo, choćby po to, by zamówić pizzę, wesprzeć moralnie i od czasu do czasu opowiedzieć coś dowcipnego. Zespół czuje się lepiej, gdy wszyscy jadą na tym samym wózku, znoszą te same niewygody i jednakowo cieszą się z sukcesu. W takim środowisku ludzie chętniej dają z siebie wszystko i bardziej skupiają się na wykonaniu zadania. Dobre kierownictwo chroni deweloperów przed naciskami z zewnątrz, robi wszystko, aby usunąć wszelkie

przeszkody, i stara się stworzyć im jak najlepsze warunki do pracy. Kierownictwo powinno być emanacją harmonii i to do niego należy dbanie o to, by w zespole panowały zdrowa atmosfera i poczucie wspólnoty.

PODSUMOWANIE

Bycie profesjonalistą i dbanie o klientów nie oznacza godzenia się na wszystko, czego żądają. Naszym zadaniem jest pomaganie klientom w ustaleniu, jakie są ich rzeczywiste potrzeby i jak mogą je zaspokoić. Profesjonalista kieruje się własną etyką i swoim kodem postępowania. Prawdziwy profesjonalista nigdy nie zrobi nic, co mogłoby zaszkodzić klientowi, mimo że klient jest gotów za to zapłacić. Klienci rzadko rozumieją specyfikę projektów informatycznych i to my mamy im uświadamiać, jak pewne decyzje mogą wpływać na tego rodzaju projekty. To my musimy powiedzieć *nie*, jeśli klient prosi o coś, co według nas nie zadziała lub czego nie da się wykonać w wyznaczonym czasie. Dokładnie tego samego oczekujemy od dobrego prawnika, księgowego czy lekarza. Nie mówimy tym profesjonalistom, co i jak mają robić. Nawet by nam nie pozwolili. Idziemy do nich z problemem (lub potrzebą) i oczekujemy, że na podstawie własnego doświadczenia powiedzą nam, jakie mamy opcje do wyboru i z jakim ryzykiem każda z nich się wiąże. Oczekujemy, że otrzymamy wystarczająco dużo informacji, aby w pełni świadomie podjąć taką czy inną decyzję. Jeśli jednak spróbujemy skłonić ich do zrobienia czegoś, co uważają za niewłaściwe lub szkodliwe, po prostu powiedzą *nie*. I my też powinniśmy tak postępować.

Skorowidz

E

F

G

I

J

K

L

M

N

O

P

Q

R

S

T

U

W

X

Z

Programuj profesjonalnie — liczą się pragmatyzm, perfekcja i cel!

Coraz więcej mówi się o dobrych praktykach programistycznych, a mimo to wciąż zdarzają się nieudane produkty. Jest wiele przyczyn tego stanu rzeczy, np. niewłaściwe zarządzanie projektami, brak wypracowanych metod rekrutacji specjalistów i kierowania zespołem. Odpowiedzią na te problemy jest wdrożenie rzemieślniczego podejścia do pracy programisty.

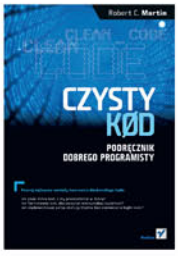
Idea ta, znana jako software craftsmanship, kładzie nacisk na profesjonalizm, techniczną perfekcję i zadowolenie klienta. Autor, współzałożyciel największej organizacji deweloperów-rzemieślników, proponuje praktyki programistyczne zaczerpnięte m.in. z metodyk programowania ekstremalnego, zwinnego i odchudzonego. Dzięki tej postawie Twój zespół może osiągnąć najwyższy poziom technicznej doskonałości tworzonych projektów!

Z książki dowiesz się, jak:

- polepszyć jakość oprogramowania dzięki podejściu rzemieślniczemu
- utrzymać wysoki poziom pisanego kodu i obsługi klienta
- być pragmatykiem — a nie dogmatykiem
- zaszczepić w zespole prawdziwy kult uczenia się

Sandro Mancuso — ekspert programistyczny z bogatym doświadczeniem w korporacjach, prelegent na konferencjach branżowych. Zwolennik idei programowania mistrzowskiego i ekstremalnego.

W serii znajdziesz też bestsellery Roberta C. Martina:



Helion
44672 numer katalogowy
księgarnia internetowa
http://helion.pl
zamówienia telefoniczne
0 801 339900
0 601 339900
Informatyka w najlepszym wydaniu

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

PRENTICE HALL

ISBN 978-83-283-2135-9
9 788328 321359
cena: 59,00 zł